

Aufgabe 1 beschreibt die Implementierung der Klasse `BigInteger`. Aufgabe 2 fokussiert die Entwicklung der Klasse `ApplicationBigInteger` mit dem entsprechenden Hauptprogramm, mit Hilfe dessen Sie die Klasse `BigInteger` testen können. Entwickeln Sie die beiden Klassen nebeneinander, d.h. implementieren Sie Ihre Klasse `ApplicationBigInteger` so weit mit, dass Sie damit die neuen Eigenschaften Ihrer Klasse `BigInteger` sukzessive testen können.

Aufgabe 1: Große Zahlen

Die feste Länge der primitiven Datentypen `int`, `long` für Ganzzahlwerte und `float`, `double` für Fließkommawerte reicht für diverse numerische Berechnungen nicht aus. Besonders wünschenswert sind beliebig große Zahlen in der Kryptographie und präzise Auflösungen in der Finanzmathematik. Für solche Anwendungen gibt es im `math`-Paket zwei Klassen: `BigInteger` für Ganzzahlen und `BigDecimal` für Gleitkommazahlen.

- a) Entwickeln Sie eine Klasse `BigInteger` zur Darstellung sehr großer positiver Zahlen. Verwenden Sie hierzu, das zur Verfügung gestellte Template. Ein `BigInteger`-Objekt speichert die Darstellung einer großen Zahl in einem `StringBuffer`. Die Klasse `StringBuffer` ermöglicht die Speicherung und Verarbeitung beliebiger Zeichenreihen. Für das Arbeiten mit dem `StringBuffer` beachten Sie bitte die Hilfsmittel in dem Template.
- b) Definieren Sie einen Konstruktor für den Wert null - `BigInteger()`
- c) Definieren Sie einen Konstruktor für den Wert einer ganzen Zahl `n` - `BigInteger(int n)`
- d) Definieren Sie einen Konstruktor, der den Wert aus einer Zeichenkette einliest – `BigInteger(String s)`
- e) Implementieren Sie eine Methode `add(BigInteger b)`. Diese liefert als Rückgabewert die Summe der Zahl `BigInteger b` und der Zahl welche durch das instantiierte Objekt der Klasse dargestellt wird. Die Methode addiert Zahlen stellenweise nach dem Schema der schriftlichen Addition, wie es den Grundschulern beigebracht wird. Falls nötig implementieren Sie weitere Methoden. Zum Auslesen der einzelnen Ziffern verwenden Sie die Methode `getDigit(int p)`.
- f) Implementieren Sie eine Methode `print()`, welche die Ausgabe einer großen Zahl auf der Konsole ermöglicht.

Aufgabe 2: Hauptprogramm

Implementieren Sie eine Klasse `ApplicationBigInteger` zur Steuerung Ihrer Applikation. Verwenden Sie hierzu das zur Verfügung gestellte Template. Zum Einlesen der großen Zahlen verwenden Sie die bereits implementierte Methode `readConsole()`

- a) Implementieren Sie im Hauptprogramm eine Methode `sumBigInteger(int pNumber)` – die Methode ruft `pNumber` – mal die Methode `readConsole()` zum Einlesen von großen Zahlen ein und gibt beim Aufruf die Summe dieser Zahlen zurück. Zum Aufaddieren der Zahlen verwenden Sie die Methode `add(BigInteger b)` der Klasse `BigInteger`.
- b) Testen Sie mit Hilfe der `main(String[] args)` Methode, die Methode `sumBigInteger(int pNumber)`. Zum Einlesen von `pNumber`, verwenden Sie die bereits vorimplementierte Abfrage. Deklarieren Sie den `BigInteger sumBI` welcher seinen Wert aus `sumBigInteger(int pNumber)` erhält. Geben Sie den Wert vom `SumBI` mit Hilfe der Methode `print()` der Klasse `BigInteger` auf der Konsole aus.
- c) Testen Sie Ihre Anwendung, für die Zahlen:
 - 5.000.000.000.000, 5.000.000.000.000
 - 123456789 und 987654321
 - 111.111.111.111, 222.222.222.222, 333.333.333.333 und 333.333.333.334

Die Ausgabe Ihres Programms sollte in etwa so aussehen:

```
Wieviele Zahlen sollen aufaddiert werden?  
2
```

```
Bitte geben Sie eine Zahl ein!  
5000000000000  
Bitte geben Sie eine Zahl ein!  
5000000000000  
Die Summe der 2 Zahlen, die Sie eingeben haben, ist:  
10000000000000
```

```
Wieviele Zahlen sollen aufaddiert werden?  
2  
Bitte geben Sie eine Zahl ein!  
123456789  
Bitte geben Sie eine Zahl ein!  
987654321  
Die Summe der 2 Zahlen, die Sie eingeben haben, ist:  
1111111110
```

```
Wieviele Zahlen sollen aufaddiert werden?  
4  
Bitte geben Sie eine Zahl ein!  
111111111111  
Bitte geben Sie eine Zahl ein!  
222222222222  
Bitte geben Sie eine Zahl ein!  
333333333333  
Bitte geben Sie eine Zahl ein!  
333333333334  
Die Summe der 4 Zahlen, die Sie eingeben haben, ist:  
1000000000000
```